5회차 · 120분

AI 기술 기초

왜 그렇게 동작하는지 알면, 무엇을 조심할지가 예측됩니다

발표자 노트 · 01

여는 말

- 이 회차부터 청중이 바뀝니다 — 실무자·개발자입니다.
- 목표는 원리 암기가 아니라 "잘하는 것과 못 하는 것을 예측하는 능력"입니다.

언어 모델이 하는 일은 정말 하나뿐입니다

지금까지의 글을 보고, 다음에 올 조각 하나를 고른다. 그리고 그걸 반복한다.

발표자 노트 · 02

정의

- 한 글자씩 답이 나타나는 걸 본 적 있는지 물어보세요 — 연출이 아니라 실제 동작입니다.
- 기대치를 낮추는 게 목적입니다. 이후 한계 설명이 자연스러워집니다.

그런데 왜 요약이 되나

요약을 시키는 게 아니라,
요약이 나올 수밖에 없는 상황을 만드는 것

(긴 글 1,000줄)

위 글의 요약:

→ 다음에 올 자연스러운 말 = 요약문

한국어: 오늘 회의는 취소되었습니다.
영어:

→ 다음에 올 자연스러운 말 = 번역문

발표자 노트 · 03

패턴 이어쓰기

- “지시를 이해했다”기보다 “그 상황에서 자연스러운 이어짐을 만들었다”에 가깝습니다.
- 이 구분이 뒤의 환각 설명을 떠받칩니다.

따라 나오는 것

무엇을 잘하고 무엇을 못 하나

잘하는 것

왜
사람 글에 흔한 패턴

예 1
요약 · 번역 · 형식 변환

예 2
초안 작성 · 분류

예 3
문장 다듬기

못 하는 것

왜
✓ 본 적 없거나 계산이 필요

예 1
✓ 정확한 계산

예 2
✓ 최신 사실 · 내부 정보

예 3
✓ 매번 동일한 출력

오른쪽 네 가지는 전부 “도구”로 해결합니다. 모델을 키워서 해결되는 게 아닙니다.

발표자 노트 · 04

능력의 경계

- 같은 질문에 답이 매번 다른 건 고장이 아니라 설계(확률적 선택)입니다.
- 다만 “같은 입력이면 반드시 같은 출력”이 필요한 일에는 안 맞는다는 뜻이기도 합니다.

환각

“모르겠습니다”는 자연스러운 이어짐이 아닙니다

질문

우리 회사 2023년 매출이 얼마였지?

답변

2023년 매출은 약 47억원입니다.

질문 뒤에 오는 자연스러운 말은 **답**입니다. 아는 게 없어도 **형태는 만들어집니다.**

발표자 노트 · 05

환각은 구조에서 나옵니다

- 버그가 아니라 성질이라 없앨 수 없고 관리해야 합니다.
- 가장 위험한 형태는 맞는 것에 틀린 것이 섞이는 경우 — 세 줄 중 두 줄이 맞으면 세 번째도 믿습니다.
- 즉석 시연이 가장 강력합니다. 없는 책을 추천해달라고 해보세요.

환각 줄이기

기억에서 만들게 하지 말고, 자료를 주고 그 안에서 답하게

위험

질문 방식
"근로기준법 연차 규정 알려줘"

무엇을 하나
기억에서 생성

출처
없음 (지어낼 수 있음)

안전

질문 방식
✓ "아래 조문에서 정리해줘" + 조문

무엇을 하나
✓ 주어진 자료에서 추출

출처
✓ 내가 준 자료

검색·DB 도구 연결도 같은 원리입니다 — 지어낼 필요를 없애는 것.

발표자 노트 · 06

대처

- “확실하지 않으면 모른다고 해줘”는 빈도를 줄이지만 없애지는 못합니다.
- 조직에서는 위험도로 나눠 높은 등급 경로에 사람 승인을 구조적으로 넣어야 합니다.

실습 · 30분

자료를 붙여보기 — 네 번 물어봅니다

1
자료 없이
우리 규정을 그냥 묻기. 숫자를 말하면 지어낸 것

2
붙이고 다시
"이 문서만 근거로. 쪽수를 붙여줘"

3
함정 질문
문서에 답이 없는 것을 묻기

4
한 문단만
잘라 붙이면 답이 흔들립니다

2단계에서 붙은 쪽수를 실제로 열어보세요. 출처가 붙었다는 것과 출처가 맞다는 것은 다릅니다.

발표자 노트 · 07

실습 진행

- 3단계가 이 실습의 핵심입니다 — 근처 조항을 끌어다 그럴듯하게 답하는 것이 가장 위험한 실패입니다.
- 그게 나오면 "명시적으로 없으면 없다고만 답해"를 넣고 다시 시켜보게 하세요. 한 줄로 잡히는 경우가 많습니다.
- 1단계에서 지어낸 화면은 캡처해두라고 하세요. 팀에 환각을 설명할 때 남의 사례보다 잘 통합니다.

"우리 데이터로 학습시키면?"

자료에 넣을까, 가중치에 넣을까

자료 붙이기(RAG)

무엇을 바꾸나
모델이 아는 것

언제 반영
문서 고치면 다음 질문부터

착수 비용
며칠~몇 주

정책이 바뀌면
한 줄 고치면 끝

파인튜닝

무엇을 바꾸나
✓ 말투 · 형식 · 판단 습관

언제 반영
✓ 재학습해야 (수 시간~수 일)

착수 비용
✓ 몇 주 + 데이터 만드는 사람 시간

정책이 바뀌면
✓ 그때까지 옛 정책을 확신에 차서 말함

파인튜닝은 사실을 집어넣는 도구가 아닙니다. 지시 → 자료 붙이기 → 도구 → 검색 구축 → 파인튜닝 순서로만 내려가세요.

발표자 노트 · 08

고르는 순서

- 파인튜닝 프로젝트가 시작되는 가장 흔한 이유는 지시를 제대로 안 써봤기 때문입니다.
- 판단 기준 하나 — 예시 다섯 개를 붙였을 때 눈에 띄게 좋아지면 그건 지시로 풀 문제입니다.
- 파인튜닝이 맞는 경우는 둘뿐입니다: 형식을 못 놓치게 고정할 때, 큰 모델 실력을 작은 모델로 옮길 때.

학습의 종류

네 가지 다 학습입니다 — 어디에 남느냐가 다릅니다

01

이번 대화 안에만

지시 — 이번 요청의 행동. 인컨텍스트 러닝이라 부르고, 대화가 끝나면 사라집니다

02

자료에

자료 붙이기(RAG) — 아는 것. 비모수적 기억이라 모델이 아니라 저장소가 배웁니다

03

도구와 절차에

도구 · 스킬 · 평가 · 가드레일 — 할 수 있는 일. 모델을 갈아타도 남습니다

04

모델 가중치에

파인튜닝 — 말투 · 형식. 사전학습 · 정렬 다음의 맨 아래 한 칸

아래로 갈수록 오래 남고, 고치기 어렵고, 비쌉니다. 되돌리기가 가장 비싼 칸을 먼저 집는 것이 가장 흔한 실수입니다.

학습의 종류

- 대조가 아니라 분류입니다. “이건 학습이고 저건 아니다”가 아니라 네 가지 다 학습이고 남는 자리가 다르다고 말해 주세요.
- 둘째 칸 — RAG 를 처음 제안한 논문이 이것을 비모수적 기억(non-parametric memory)이라 부릅니다. 가중치 안에 넣는 대신 밖에 두고 꺼내는 기억이라, 문서를 하나 넣으면 그다음 질문부터 아는 것이 늘어납니다.
- 셋째 칸이 조직에서 실제로 가장 많이 쌓이는데 부르는 이름이 아직 안 굳었습니다 — 컨텍스트 엔지니어링 · 하네스 엔지니어링 같은 말이 돌아다닙니다. 앞의 둘이 “무엇을 아는가”를 늘린다면 이쪽은 “무엇을 할 수 있는가”를 늘리고, 모델을 갈아타도 그대로 남습니다.
- 넷째 칸의 셋 중 사전학습과 정렬은 모델 만드는 회사가 이미 끝냈습니다. “우리 데이터로 학습시키자”가 뜻하는 것은 파인튜닝 한 칸이고, 그만큼 바꿀 수 있는 것도 얇습니다.
- 넷을 굳이 나누는 실무적인 이유는 틀렸을 때 치르는 값입니다 — 지시는 문장을 고치면 끝, 자료는 문서 한 줄, 도구는 설정, 파인튜닝은 재학습 전까지 확신에 차서 계속 틀립니다.

도구 사용

모델에 손을 달아주기

1
사람이 묻는다
A-1023 재고
얼마나 남았어?

2
모델이 판단
이건 내가 모르는 값이다

3
호출 요청
재고조회("A-1023")

4
시스템이 실행
모델이 아니라
시스템이 합니다

5
결과로 답
현재 47개입니다

4단계가 핵심입니다 — 실행은 시스템이 하므로 그 사이에 사람 승인을 끼울 수 있습니다.

발표자 노트 · 10

도구 구조

- 답에 근거가 생기면 검증할 수 있습니다. 환각이 크게 줄어드는 이유입니다.
- 읽기 도구부터 붙이세요. 쓰기(발송·결제·삭제)는 승인 설계와 함께입니다.

에이전트

도구를 한 번 쓰는 것과 스스로 반복하는 것

도구 사용

반복
1회

사람 개입
매 턴

실패 양상
한 번 틀리면 한 번

사람의 일
질문하고 답 보기

에이전트

반복
✓ 목표 달성까지

사람 개입
✓ 시작과 끝

실패 양상
✓ 틀린 전제 위에 계속 쌓음

사람의 일
✓ 목표를 정확히 주고 멈출 지점 정하기

에이전트는 틀린 방향으로도 열심히 갑니다. 그래서 사람의 일이 앞으로 이동합니다.

발표자 노트 · 11

에이전트

- 매출 분석 5회전 예시를 말로 풀어주세요 — 3회전의 "원인을 봐야겠다"가 스스로 내린 판단입니다.
- 자율 루프에는 바깥에서 한계를 겁니다 — 반복 횟수·비용·시간·승인 지점.

컨텍스트

모델은 아무것도 기억하지 않습니다

01

매번 다시 넣는 것

대화가 이어지는 건 지난 대화를 통째로 다시 보내기 때문

02

창에는 크기가 있음

차면 오래된 것부터 밀려납니다 — 대개 앞쪽에 중요한 게 있습니다

03

뒤로 갈수록 비쌈

20번째 요청 = 앞의 19턴 전부 + 새 질문

"모델이 멍청해진 것 같다"의 상당수가 이것입니다. 모델은 그대로이고 보고 있는 것이 달라졌습니다.

발표자 노트 · 12

컨텍스트

- 증상을 나열해주세요 — 말투가 달라짐, 하지 말라는 걸 함, 이미 정한 걸 다시 물음.
- 에이전트에서 더 심합니다. 도구 결과가 매 회전 쌓입니다.
- 대처: 작업 쪼개기 · 제약 재확인 · 도구 결과 요약 · 정리 후 새 대화.

기억

창을 닫으면 남는 것이 없습니다

- 1층 — 지금 보고 있는 대화 전체. 창을 닫으면 사라집니다
- 2층 — 앞부분을 압축한 요약. 이것도 창 안에 있어 함께 사라집니다
- 3층 — 대화 밖 파일에 적어둔 것. 지우기 전까지 남습니다
- 밀려나기 전에도 나빠집니다 — 끝난 중간 단계가 쌓이면 품질이 조용히 떨어집니다

메모리에 자격증명을 적지 마세요. 한 번 적히면 이후 모든 세션의 컨텍스트로 다시 읽힙니다.

발표자 노트 · 13

세션을 넘는 기억

- MIT NANDA 보고서가 95% 실패의 근본 원인으로 지목한 것이 “배포된 시스템에 기억이 없다”였습니다.
- 2층과 3층을 헷갈리면 “요약했으니 기억하겠지” 하다가 다음 날 다시 설명하게 됩니다.
- 컨텍스트 로트의 신호 — 정한 규칙을 슬며시 어김, 이미 확인한 것을 다시 확인, 답이 일반론으로.
- 좋은 메모리는 큰 것이 아니라 관리되는 것입니다. 지우는 절차가 없으면 6개월 뒤 부채가 됩니다.

좋은 지시

네 가지 요소

01

목표

빠지면 엉뚱한 것을 만듦. “무엇이 문제다”로 적으면 명확해집니다

02

범위

빠지면 건드리면 안 될 곳을 건드림. 하지 말 것으로 적으세요

03

완료 조건

빠지면 판정이 갈림. 판정이 수동이면 자동화 이득이 깎입니다

04

맥락

빠지면 바퀴를 다시 발명. 이미 실패한 것도 알려주세요

일회성 질문에는 과합니다. 반복 작업과 에이전트에 맡기는 작업에 씁니다.

발표자 노트 · 14

네 요소

- 가장 자주 빠뜨리는 건 맥락입니다 — 내가 아는 걸 상대도 안다고 가정하기 때문.
- 완료 조건을 못 정하면 그 작업은 아직 맡길 준비가 안 된 것입니다.

실습 · 30분

나쁜 지시 다섯 개를 직접 고쳐봅니다

나쁜 지시

보고서
지난달 매출 보고서 만들어줘

메일
거래처에 사과 메일 써줘

정리
이 데이터 정리해줘

무인 작업
매일 아침 매출 슬랙에 올려줘

빠진 요소

보고서
✓ 맥락 — B제품 단종을 안 주면 이상 징후로 해석

메일
✓ 범위 — 보상 언급 금지. 안 적으면 약속이 됩니다

정리
✓ 완료 조건 — “추측해서 채우지 말 것”

무인 작업
✓ 실패 시 동작 — 틀린 숫자보다 안 올리는 게 낫습니다

사람이 중간에 보지 않는 작업은 “잘 됐을 때”가 아니라 “안 됐을 때”를 적어야 합니다.

발표자 노트 · 15

지시 고쳐쓰기

- 답을 보여주기 전에 각자 종이에 먼저 고쳐 쓰게 하세요. 20분이면 됩니다.
- 매번 같은 요소를 빠뜨린다면 그게 그 사람의 습관입니다 — 가장 흔한 것은 맥락입니다.
- 네 번째 줄에서 난이도가 올라갑니다. 되돌릴 수 없는 동작(슬랙 게시)과 데이터 갱신 시점을 같이 짚으세요.

평가(EVALS)

개인은 눈으로 판정하지만 조직은 못 합니다

개인

누가 판정
본인

언제
즉시

기준
자기 감

하루 건수
몇 건

조직

누가 판정
✓ 매번 다른 사람

언제
✓ 나중에, 또는 안 함

기준
✓ 사람마다 다름

하루 건수
✓ 수백~수천

수백 건을 사람이 다 보면 자동화의 이득이 사라집니다. 그래서 판정 자체를 자동화합니다.

발표자 노트 · 16

평가가 필요한 이유

- 평가가 없으면 “프롬프트를 고쳤더니 이건 좋아졌는데 저건 나빠졌다”를 아무도 모릅니다.
- 소프트웨어에서 이미 겪은 문제이고 답도 같습니다 — 회귀 테스트.

평가 시작하기

케이스 20개면 시작할 수 있습니다

- 1주차 — 실제 케이스 20개를 스프레드시트에 (입력 / 기대 결과)
- 2주차 — 돌려서 몇 개 맞는지 센다. 이게 기준선
- 3주차 — 프롬프트를 고칠 때마다 다시 돌린다
- 이후 — 틀린 케이스가 나오면 데이터셋에 추가

AI 출력은 매번 조금씩 다릅니다. “100% 통과”가 아니라 “기준선보다 나아졌는가”가 판정입니다.

발표자 노트 · 17

골든 데이터셋

- 반드시 넣을 것: 흔한 케이스 · 실제로 틀렸던 케이스 · 경계 · 하면 안 되는 것.
- 두 번째가 핵심입니다 — 데이터셋이 시간이 갈수록 좋아집니다.
- LLM-as-judge를 쓰면 채점기도 가끔 사람이 표본 검사해야 합니다.

모델 고르기

제일 좋은 것 하나로 다 돌리면

- 분류 · 태깅 · 대량 반복 → 경량
- 대부분의 일상 업무 → 중급
- 검수 · 최종 판정 · 복잡한 판단 → 상급
- 두 개로 시작하세요 — 일상용 하나, 검수용 하나

5배

경량으로 될 일에 내는 값

상급 · 중급 · 경량은 성능과 가격이 함께 움직입니다

모델과 비용

- 출력이 입력보다 5배 비싸고, 매 턴마다 "지금까지 전부"가 입력으로 다시 계산됩니다.
- 아끼는 법 셋 — 같은 앞부분은 캐시, 생각 깊이를 낮추기, 넣는 것을 줄이기.
- 지시문 맨 앞에 오늘 날짜를 넣으면 캐시가 매일 깨집니다. 고정된 것 먼저, 바뀌는 것 뒤로.
- 큰 컨텍스트 창은 여유이지 목표가 아닙니다 — 채울수록 비싸고 느리고 정확도가 떨어집니다.

실습 · 30분

컨텍스트를 일부러 넘겨보기

1
표식 심기
대화 맨 앞에 규칙 둘 — 형식(이니셜)과 사실(코드명)

2
길게 채우기
긴 글을 5~8회 나눠 붙이며 정리시킵니다

3
되묻기
규칙을 다시 알려주지 말고 물어봅니다

4
되살리기
되묻기 · 제약 다시 적기 · 요약 후 새 대화 — 무엇이 듣는지

가장 위험한 결과는 “모르겠습니다”가 아니라 **다른 코드명을 자신 있게 말하는 것**입니다. 모델은 잊었다고 말하지 않고 채웁니다.

실습 진행

- 2단계가 지루한 게 정상입니다 — 실무에서도 컨텍스트는 지루하게 찹니다.
- 형식 규칙(이니셜)이 사실 규칙(코드명)보다 먼저 무너지는 경우가 많습니다. 어느 쪽이 먼저인지 물어보세요.
- 방법 A(되묻기)가 안 듣는 것을 반드시 확인시키세요. 이걸 봐야 안 하게 됩니다.
- 끝까지 안 무너지면 그 모델의 창이 넓은 것이고, 그것도 결과입니다 — 대신 비용과 속도를 보게 하세요.

정리

오늘 가져가실 것

1. 모델은 다음 말을 맞히는 기계 — 그래서 잘하는 것과 못 하는 것이 예측됩니다
2. 환각은 구조에서 나옵니다. 자료를 주고 그 안에서 답하게 하세요
3. 사실은 찾아 붙이고, 말투와 형식만 학습으로 고정합니다
4. 실행은 시스템이 합니다 — 그 사이가 승인을 넣을 자리입니다
5. 창을 닫으면 사라집니다. 남길 것은 대화 밖 파일에
6. 조직에서는 판정도 자동화해야 합니다 (Evals)
7. 대량은 싼 모델로, 검수만 비싼 모델로

발표자 노트 · 20

정리

- 다음 회차는 도구를 실제로 연결하는 표준(MCP)입니다.
- 이 회차는 개념이 많아 지칩니다. 마지막에 “오늘 것 중 하나만 기억한다면”을 물어보고 닫으세요.
- 질문이 많이 남았으면 용어집 덱을 5분 띄워 정리하고 넘어가는 것도 방법입니다.